Amendments to the Specification:

Please replace the paragraph beginning at page 1, line 6 with the following rewritten paragraph:

This application is a divisional of U.S. Serial No. 09/747,056 filed December 22, 2000 and ~~The present invention~~ claims the benefit of U.S. Provisional Application Serial No. 60/171,911 ~~entitled "Methods and Apparatus for Loading a Very Long Instruction /word Memory" and~~ filed December 23, 1999 which is incorporated by reference herein in its entirety.

Please replace the paragraph beginning at page 3, line 16, with the following rewritten paragraph.

The present invention may be applicable to a variety of processing and array designs; however, an exemplary and presently preferred architecture for use in conjunction with the present invention is the ManArray™ architecture. Further details of a presently preferred ManArray core, architecture, and instructions for use in conjunction with the present invention are found in U.S. Patent Application Serial No. 08/885,310 filed June 30, 1997, now U.S. Patent No. 6,023,753, U.S. Patent Application Serial No. 08/949,122 filed October 10, 1997, now U.S. Patent No. 6,167,502, U.S. Patent Application Serial No. 09/169,255 filed October 9, 1998, now U.S. Patent No. 6,343,356, U.S. Patent Application Serial No. 09/169,256 filed October 9, 1998, now U. S. Patent No. 6,167,501, U.S. Patent Application Serial No. 09/169,072, filed October 9, 1998, now U.S. Patent No. 6,219,776, U.S. Patent Application Serial No. 09/187,539 filed November 6, 1998, now U.S. Patent No. 6,151,668, U.S. Patent Application Serial No. 09/205,55~~8~~8 filed December 4, 1998, now U.S. Patent No. 6,173,389, U.S. Patent Application Serial No. 09/215,081 filed December 18, 1998, now U.S. Patent No. 6,101,592, U.S. Patent

Application Serial No. 09/228,374 filed January 12, 1999 now U.S. Patent No. 6,216,223, ~~and entitled "Methods and Apparatus to Dynamically Reconfigure the Instruction Pipeline of an Indirect Very Long Instruction Word Scalable Processor",~~ U.S. Patent Application Serial No. 09/238,446 filed January 28, 1999, now U.S. Patent No. 6,366,999, U.S. Patent Application Serial No. 09/267,570 filed March 12, 1999, now U.S. Patent No. 6,446,190, U.S. Patent Application Serial No. 09/337,839 filed June 22, 1999, U.S. Patent Application Serial No. 09/350,191 filed July 9, 1999, now U.S. Patent No. 6,356,994, U.S. Patent Application Serial No. 09/422,015 filed October 21, 1999 ~~entitled "Methods and Apparatus for Abbreviated Instruction and Configurable Processor Architecture",~~ now U.S. Patent No. 6,408,382, U.S. Patent Application Serial No. 09/432,705 filed November 2, 1999 entitled "Methods and Apparatus for Improved Motion Estimation for Video Encoding", U.S. Patent Application Serial No. 09/471,217 filed December 23, 1999 entitled "Methods and Apparatus for Providing Data Transfer Control", U.S. Patent Application Serial No. 09/472,372 filed December 23, 1999 ~~entitled "Methods and Apparatus for Providing Direct Memory Access Control"~~, now U.S. Patent No. 6,256,683, U.S. Patent Application Serial No. 09/596,103 ~~entitled "Methods and Apparatus for Data Dependent Address Operations and Efficient Variable Length Code Decoding in a VLIW Processor"~~ filed June 16, 2000, now U.S. Patent No. 6,397,324, U.S. Patent Application Serial No. 09/598,567 entitled "Methods and Apparatus for Improved Efficiency in Pipeline Simulation and Emulation" filed June 21, 2000, U.S. Patent Application Serial No. 09/598,564 ~~entitled "Methods and Apparatus for Initiating and Resynchronizing Multi-Cycle SIMD Instructions"~~ filed June 21, 2000, now U.S. Patent No. 6,622,238, U.S. Patent Application Serial No. 09/598,566 entitled "Methods and Apparatus for Generalized Event Detection and Action Specification in a Processor" filed June 21, 2000, and U.S. Patent Application Serial No.

09/598,084 ~~entitled "Methods and Apparatus for Establishing Port Priority Functions in a VLIW Processor"~~ filed June 21, 2000, now U.S. Patent No. 6,654,870, U.S. Patent Application Serial No. 09/599,980 entitled "Methods and Apparatus for Parallel Processing Utilizing a Manifold Array (ManArray) Architecture and Instruction Syntax" filed June 22, 2000, ~~as well as, Provisional Application Serial No. 60/113,637 entitled "Methods and Apparatus for Providing Direct Memory Access (DMA) Engine" filed December 23, 1998, Provisional Application Serial No. 60/113,555 entitled "Methods and Apparatus Providing Transfer Control" filed December 23, 1998, Provisional Application Serial No. 60/139,946 entitled "Methods and Apparatus for Data Dependent Address Operations and Efficient Variable Length Code Decoding in a VLIW Processor" filed June 18, 1999, Provisional Application Serial No. 60/140,245 entitled "Methods and Apparatus for Generalized Event Detection and Action Specification in a Processor" filed June 21, 1999, Provisional Application Serial No. 60/140,163 entitled "Methods and Apparatus for Improved Efficiency in Pipeline Simulation and Emulation" filed June 21, 1999, Provisional Application Serial No. 60/140,162 entitled "Methods and Apparatus for Initiating and Re-Synchronizing Multi-Cycle SIMD Instructions" filed June 21, 1999, Provisional Application Serial No. 60/140,244 entitled "Methods and Apparatus for Providing One-By-One Manifold Array (1x1 ManArray) Program Context Control" filed June 21, 1999, Provisional Application Serial No. 60/140,325 entitled "Methods and Apparatus for Establishing Port Priority Function in a VLIW Processor" filed June 21, 1999, Provisional Application Serial No. 60/140,425 entitled "Methods and Apparatus for Parallel Processing Utilizing a Manifold Array (ManArray) Architecture and Instruction Syntax" filed June 22, 1999, Provisional Application Serial No. 60/165,337 entitled "Efficient Cosine Transform Implementations on the ManArray Architecture" filed November 12, 1999, and Provisional Application Serial No. 60/171,911~~

~~entitled "Methods and Apparatus for DMA Loading of Very Long Instruction Word Memory" filed December 23, 1999, Provisional Application Serial No. 60/184,668 entitled "Methods and Apparatus for Providing Bit-Reversal and Multicast Functions Utilizing DMA Controller" filed February 24, 2000, Provisional Application Serial No. 60/184,529 entitled "Methods and Apparatus for Scalable Array Processor Interrupt Detection and Response" filed February 24, 2000, Provisional Application Serial No. 60/184,560 entitled "Methods and Apparatus for Flexible Strength Coprocessing Interface" filed February 24, 2000, Provisional Application Serial No. 60/203,629 entitled "Methods and Apparatus for Power Control in a Scalable Array of Processor Elements" filed May 12, 2000, and Provisional Application Serial No. 60/241,940 entitled "Methods and Apparatus for Efficient Vocoder Implementations" filed October 20, 2000,~~ all of which are assigned to the assignee of the present invention and incorporated by reference herein in their entirety.

Please replace the paragraph beginning at page 19, line 1, with the following rewritten paragraph:

When this type-~~1~~2 VIM DMA operation begins, the "data" packet is transferred beginning with the LV2 "data" item over the 32-bit DMA bus 915 and stored into DMA register 1 (DR1) 910. The output 917 of DR1 is provided to VIM controller 922 which provides "cycle borrow" control to the DMA interface. The received LV2 instruction causes the correct Vx base address register to be selected via path 926 and adder 924 adds the base Vx value to the offset found in the received LV2 unit VIM offset field 945 (Fig. 4C bits 7-0 for the ALU defined by bits 17-15). The resultant sum is stored in register 930 for later use. A programmer can use the two Vx base address registers to aid this process where one, say register V0, can be set up for the instruction

iVLIW usage and the other, register V1, for DMA base address use. The DMA packet will continue sending the next LV2 SIWs one at a time through DR1 910 to be loaded into the ALU VIM portion. For each SIW received at 910, the SIW is loaded directly into the ALU VIM portion under control of the VIM control 922 at the LV2 instruction specified address for the first SIW and then incrementally as long as the instruction side does not take priority. The DMA VIM control logic automatically increments the VIM address in preparation for the next SIW received on the DMA interface. The VIM controller 922 selects multiplexer 936 for the DMA VIM address 929 and selects the multiplexer 954 to the DR1 register output 917 and in a single cycle the SIW is then loaded into the ALU VIM 903. The DMA interface is then allowed to proceed with the next SIW "data" item of the DMA packet if there is one. It is noted that if the "data" packet is common to multiple PEs, the multiple PEs' functional VIM portions can be loaded in parallel and in synchronism. The DMA operation continues for each functional partition of the full Type-2 VIM apparatus.